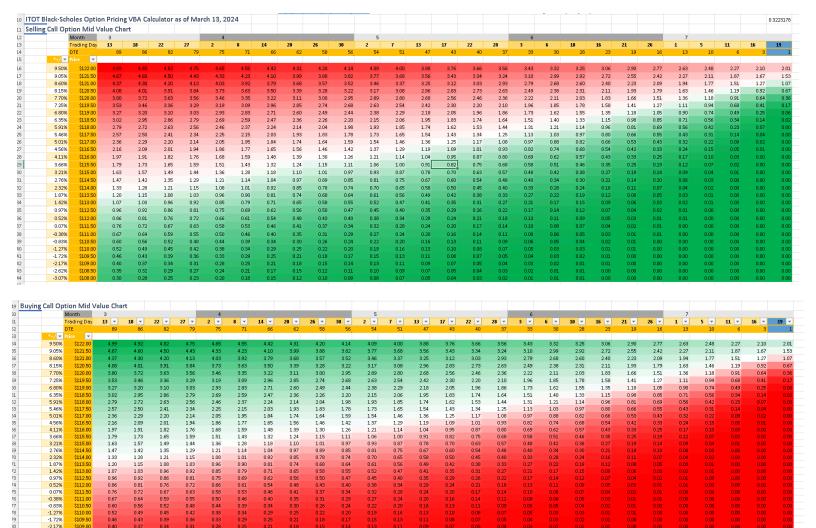
Black-Scholes Options Pricing Model

By Bertan Jonjon Kurtsoy

Overview

This document highlights the process of creating the Black-Scholes option pricing model for buy and sell call contract of ITOT(iShares Core S&P Total US Stock Market ETF) as of March 13, 2024, expiring on July 19, 2024. Final Result below, sell(top) and buy(bottom).



Tools & skills used

- Microsoft Excel
- VBA(Visual Basic for Applications) to create specific function
 - o "DaysUntilExpiration(date(yyyy,mm,dd), date(2024,7,19))"
 - o "BlackScholesCallPrice(UnderlyingPrice, DaysUntilExpiration)"
 - o "BlackScholesPutPrice(UnderlyingPrice, DaysUntilExpiration)" Put contracts were excluded
- Yahoo Finance for sourcing historic price data of ITOT
 - o Calculated standard daily deviation YTD; further used for volatility parameter
- Option Contract Parameters
 - o Underlying strike price
 - o Strike price
 - Volatility
 - o Interest Rate (10 yr Treasury rate)
 - o Dividend yield
 - o Time Until Expiration

Project Timeline: Processes, Challenges, and Solutions(chronological order)

Here I input the historical price data 0/2/24 to 0/3/24(cropped out) from Yahoo Finance which was used to find the standard daily deviation 'year to date' of ITOT. This would be used to calculate parameters such as volatility.

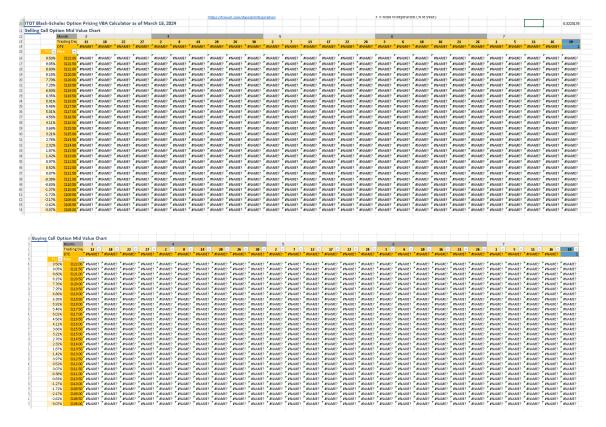
AJ	AK	AL	AM	AN
Open/Close, %Change, YTD				
Date	Open Pri	Close Pric	%Chang	Standard Daily Deviation YT
2-Jan-24	104.47	104.51		0.007451578
3-Jan-24	103.97	103.43	-1.03%	
4-Jan-24	103.35	103.16	-0.26%	
5-Jan-24	103.14	103.32	0.16%	
8-Jan-24	103.41	104.78	1.41%	
9-Jan-24	104.18	104.59	-0.18%	
*******	104.67	105.13	0.52%	
11-Jan-24	105.28	105.05	-0.08%	
*******	105.41	105.09	0.04%	
*******	104.71	104.6	-0.47%	
*******	103.9	104.03	-0.54%	
*******	104.43	104.88	0.82%	
*******	105.2	106.22	1.28%	
*******	106.58	106.61	0.37%	
*******	106.77	106.84	0.22%	
*******	107.51	106.83	-0.01%	
*******	107.36	107.39	0.52%	
*******	107.34	107.31	-0.07%	
*******	107.39	108.32	0.94%	
*******	108.11	108.11	-0.19%	
*******	107.66	106.33	-1.65%	
1-Feb-24	106.65	107.7	1.29%	
2-Feb-24	107.68	108.7	0.93%	
5-Feb-24	108.49	108.21	-0.45%	
6-Feb-24	108.34	108.51	0.28%	
7-Feb-24	109.07	109.37	0.79%	
8-Feb-24	109.39	109.65	0.26%	
9-Feb-24	109.76	110.33	0.62%	
*******	110.33	110.42	0.08%	
*******	108.84	108.66	-1.59%	
*******	109.34	109.88	1.12%	
*******	110.13	110.73	0.77%	
*******	110.66	110.14	-0.53%	
*******	109.7	109.44	-0.64%	

Parameters - 3/13/24					
Underlying Price S	113.6				
strike price K	120				
volatility σ	14.24%				
Interest rate r	4.19%				
Dividend yld q	1.70%				
DTE	90				
Time to exp t	24.66%				
In(S/K)	-0.0548				
t(r-q+σ^2/2)	0.0086				
σ*sqrt(t)	0.0707				
d_1	-0.6531				
d_2	-0.7238				
$N(d_1)$	0.2568				
$N(d_2)$	0.2346				
$N(-d_1)$	0.7432				
N(-d2)	0.7654				
e ^{-rt}	0.9897				
Xe ^{-rt}	118.7666				
e ^{-qt}	0.9958				
Se ^{rqt}	113.1248				

I then found the following option contract parameters based on the Black-Scholes equation.

Includes the following:

- o Underlying strike price
- o Strike price
- Volatility
- o Interest rate(10 year Treasury rate)
- o Dividend yield
- Time Until Expiration



In the images above, I created the charts above representing sell(top) and buy(bottom) call values given (+/-) % change in stock price and the trading day. However, I ran into two issues which I detail below.

```
Main-Robinhood-Investments-Portfolio(AutoRecovered)(AutoRecovered)xixx - Module1 (Code)

jeneral)

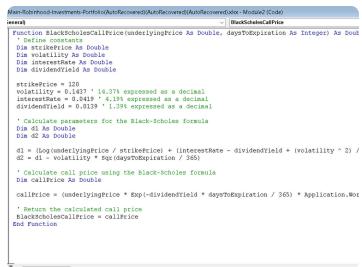
Function daysUntilExpiration (currentDate As Date, expirationDate As Date) As Int
Dim tradingDays As Integer
Dim holidays As Variant

Define the list of holidays
holidays = Array(#1/1/2024#, #1/15/2024#, #2/19/2024#, #4/19/2024#, #5/27/2024#

Count the number of trading days between current date and expiration date
Do While currentDate < expirationDate
If Weekday(currentDate) <> vbSaturday And Weekday(currentDate) <> vbSunday Ther
For i = LBound(holidays) To UBound(holidays)
If currentDate = holidays(i) Then
Exit For
End If
Next i
If i > UBound(holidays) Then
tradingDays = tradingDays + 1
End If
End If
CurrentDate = currentDate + 1
Loop

daysUntilExpiration = tradingDays
End Function
```

Issue 1: The trading days along the y-axis of the chart was not a unusable parameter, thus instead of manually doing the math for what each trading day had left in days until expiration, I opted to create a VBA module that would take the trading days and reference the official 2024 Trading Day Calendar(to avoid counting holidays and weekends) to give days until expiration.



Issue 2: Given that I attempted to squeeze the entire Black-Scholes equation into each cell which crashed my computer every time I refreshed the excel. I concluded that a VBA module for the Black-Scholes call price as an excel function would run better on my computer.

Final Result:

After comparing my values to Robinhood's Buy/sell option contract values of ITOT, I conclude that my option pricing model is accurate to that of Robinhood and presumably other brockerages.

